

**MULTI-STYLE WEB CONTENT RESTRUCTURING
FOR EFFICIENT MOBILE WEB BROWSING**

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Xiaojun Xia

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

April 2011

Fargo, North Dakota

North Dakota State University
Graduate School

Title

MULTI-STYLE WEB CONTENT RESTRUCTURING

FOR EFFICIENT MOBILE WEB BROWSING

By

XIAOJUN XIA

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

North Dakota State University Libraries Addendum

To protect the privacy of individuals associated with the document, signatures have been removed from the digital version of this document.

ABSTRACT

Xia, Xiaojun, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, April 2010. Multi-Style Web Content Restructuring for Efficient Mobile Web Browsing. Major Professor: Dr. Jun Kong.

Mobile devices such as smart phones, PDAs, and tablets are getting more and more popular every day. As a result, more people tend to use mobile devices to access Web contents. Therefore, mobile browsing becomes an important means of Internet access for daily life. In order to have smooth and user-friendly Web browsing experiences, Web designers should consider the limitations of mobile devices, such as a small screen size. However, most Web sites do not provide a mobile version, which makes it difficult to browse the contents on the small screens of mobile devices. This study presents a novel approach that restructures the important information from a full Web page to fit for mobile browsing. Unlike existing adaptation approaches, my work supports user customization on the information restructuring. In other words, the adaption engine can restructure information according to a user's personal favors.

My approach consists of two sub-systems. The first sub-system is responsible to discover the organization of information underlying a Web page, and the second sub-system receives the structured information and uses a user-friendly interface to present the information. My approach also provides users with some customization options in order to improve the user experience. The first sub-system is out of the scope of this paper. This paper mainly focuses on the second sub-system. In order to produce a user-friendly interface, firstly I analyzed existing mobile Web sites,

which are specifically designed for mobile devices. In this analysis, I summarized the patterns and the functionalities of these mobile sites in order to customize the design to support a wide range of different Web pages. For these purposes, I have designed an adaptive layout that the users can customize how the information will appear on the mobile device screen easily.

Keywords: Mobile Web Browsing, Multi-Style, Adaptive Layout.

ACKNOWLEDGMENTS

This research project would have not been possible without the support of many people. The author wishes to express his gratitude to his supervisor, Dr. Jun Kong, who has offered invaluable assistance, support and guidance.

Deepest gratitude is also due to the members of the graduate committee: Dr. Changhui Yan, Dr. Jing Shi, and Dr. Wei Jin, for without their suggestions and assistance this study would not have been successful.

The author wishes to express his love and gratitude to his family for their unconditional understanding, support and endless love through the duration of this study.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS.....	v
LIST OF TABLES.....	viii
LIST OF FIGURES	ix
CHAPTER 1. INTRODUCTION.....	1
CHAPTER 2. RELATED WORK.....	4
2.1 Server Side Approaches	5
2.1.1 Mobile specific markup languages	5
2.1.2 Automatic mobile Web site creating tools.....	6
2.2 Client Side Navigation Techniques.....	7
2.3 Web Content Restructuring	9
2.3.1 Approaches of client-side restructuring	9
2.3.2 Approaches of server-side restructuring	10
2.3.3 Comparing with my approach.....	11
CHAPTER 3. EVALUATION OF MOBILE WEBSITES AND MOBILE WEBSITE GUIDELINES.....	12
3.1 Navigation Facility.....	13
3.2 Information Presentation	18
3.3 Evaluation Analysis.....	19
3.4 Guidelines for Mobile Web Design & Development.....	21
CHAPTER 4. RESTRUCTURING	25

4.1 Multi-style Information Presentation	26
4.2 Module Design.....	30
4.3 Class Design	33
4.4 Class Diagram.....	35
CHAPTER 5. CONCLUSION AND FUTURE WORK	40
5.1 Conclusion	40
5.2 Future Work.....	41
REFERENCES	42

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Evaluation of Navigation Facility	17
2. Evaluation of Information Presentation.....	20

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. 2 Level Navigation Menu of cnn.com	14
2. 3 Level Navigation Menu of amazon.com.....	14
3. Table of Contents of Fox news and Staples.....	15
4. Navigation lists of m.yellowpage.com and mobile.dominos.com	16
5. Collapsing Layout in YouTube.com	19
6. Display Option for Table of Contents	26
7. Layout Options for Detailed Page	27
8. Anchor Link Layout Screenshot	29
9. Single Page Layout Screenshot.....	30
10. Collapsing Layout Screenshot.....	31
11. The Prototype Architecture.....	32
12. The Activity Diagram of AsyncTask.....	36
13. The Activity Diagram of PhotoLoader Class	37
14. Class Diagram	38

CHAPTER 1. INTRODUCTION

Advanced mobile devices such as smart phones, personal digital assistants are getting more and more popular every day. These mobile devices are normally equipped with a wireless network connection and a Web browser through which users can access the Internet from anywhere and at any time. As a result, more and more users tend to browse the Web using their mobile devices. A large number of users use the mobile devices to read news stories, find production information, search nearby restaurants, or even online shopping. The mobile access to Web information, however, raises a new challenge to the Web page presentation. The small screen of mobile devices is one of the major reasons why many users still hesitate to use such devices to access the Internet. Users have to zoom in/out & scroll both vertically and horizontally to find the information of interest since most of the Web pages are designed for desktop, which is equipped with large display. Those Web pages are referred to as full Web pages in this paper.

Without an adaptation, it is frustrating to browse those pages on mobile devices because of the limitations of mobile devices. Hence, it is necessary to provide a more user-friendly way of mobile Web browsing. Markup languages, such as WML and XHTML Mobile Profile (XHTML-MP), have been proposed to support the rendering of information on mobile devices. Though the number of Web pages in the form of WML or XHTML is growing fast, Web pages tailored to personal computers still dominate the Internet. Furthermore, keeping two versions of presentations, one for desktops and the other for mobile devices, raises the

maintenance cost and causes inconsistency. Therefore, it is desirable to automatically adapt a Web page from a desktop presentation to a mobile presentation.

In order to support effective browsing of desktop presentations on mobile devices, the first challenging issue is to recognize important information from a full Web page. After recognizing major information blocks, the second challenging issue is to restructure the recognized information to fit the mobile browsing. This paper focuses on the second challenge. To restructure the Web content on mobile devices, all existing approaches support only one fixed style (e.g. a long-column style or a multi-page style) to render an adaptive layout for a diverse range of users, Web site styles, and devices. However this “one size fits all” strategy could not offer universal usability. Even regardless of users’ preferences, one fixed style could not cater for the proliferation of diverse types of mobile devices with different display and computing capacities. For example, a thumbnail-based adaptive style cannot fit on an old cellular phone, which does not support graphics. Therefore, it is desirable to provide a layout mechanism, adaptive to the user’s personal preferences, the requirements of the device, and information being accessed.

The contributions of this paper can be summarized as the following:

- I summarized a set of popular layouts employed in existing mobile Websites through evaluating 18 existing mobile Websites. This evaluation

gives a solid basis to compare different approaches and provides guidance on how to design an adaptive layout.

- I proposed a novel multi-style Web content restructuring approach for efficient mobile browsing. My approach provides an efficient navigation facility and much better usability comparing to existing mobile Web browsers.
- I have implemented a prototype, which provides customization capabilities to users. The prototype enables users to choose a presentation style that fits their needs on mobile devices. Based on evaluation results, the prototype provides good user experiences on browsing Web pages and compared with normal phone browsers, users are more satisfied with my prototype experience.

The rest of this paper is organized as follows: Chapter 2 describes the related technologies and background knowledge of Web content restructuring for mobile Web browsing. Chapter 3 presents a comprehensive classification of adaptive layouts through an evaluation of 18 existing mobile Websites. Chapter 4 introduces the system design including system overview, detailed design and implementation. Chapter 5 concludes the study and presents future work.

CHAPTER 2. RELATED WORK

Browsing desktop-size Web pages on small screen devices has been widely discussed and many solutions [Opera, Bitst, Gom11, Skwee, Mac03, Bau04, Bjo99, Lam05, Mil03, Wob02] have been proposed. Based on the approaches employed in these solutions, these solutions can be classified into three different categories:

- **Server Side Approaches.** In this category some approaches proposed to use mobile specific markup languages such as WML and XHTML Mobile Profile (XHTML-MP) to create a mobile version Website dedicated to mobile browsing, while others proposed many interactive tools with which users can easily create mobile optimized Web site. Instead of using mobile specific markup languages, these new mobile Web sites are built with the techniques for desktop version Websites but supported by mobile browsers.
- **Client Side Navigation Techniques.** Approaches in this category display Web pages at a scale that works for most Web contents originally designed for the desktop. These approaches do not change the presentation of the original Web pages. By providing an overview of the Web contents initially, these approaches support many user-friendly interaction mechanisms such as zoom in/out, thumbnail to facilitate mobile users find the contents they are interested in.
- **Web Content Restructuring.** Approaches in this category analyze a desktop Web page and then automatically create a mobile presentation that fits the small screen of a mobile device and places related information in proximity.

Differencing from those sever side approaches, these approaches do not require a server or server application. The restructuring job is done dynamically on the client side while the mobile users browse the Web sites. In these approaches, the generated adaptive mobile presentation is normally different from the original Web page. These approaches in general are generic to be applied to different Web.

2.1 Server Side Approaches

Sever side approaches indicate the technologies that created mobile optimized Web sites. These technologies can be further divided into two sub categories: mobile specific markup languages and automatic mobile Web site converting tools.

2.1.1 Mobile specific markup languages

In the last decade, many mobile specific markup languages, such as HDML [W3c97], WML and XHTML Mobile Profile (XHTML-MP), have been proposed to support the rendering of information on mobile devices. Based on extensive experiences of mobile Web browsing, these mobile specific markup languages can be employed to create mobile optimized Web sites. Those languages are specifically designed to fit the unique features of mobile devices (such as screen size, input model and computing capability).

The Handheld Device Markup Language (HDML) [W3c97] is originally developed by Unwired Planet in 1996. HDML is a markup language intended for display on handheld computers, information appliances, Smartphone, etc. It is

similar to HTML but for wireless and handheld devices with smaller displays like PDA, smart phones and so on. HDML was submitted to W3C for standardization, but was not turned into a standard.

Wireless Markup Language is another markup language intended for devices that implement the Wireless Application Protocol (WAP) specification. WML 1.1 standard was created in 1998 and WML 2.0 was specified in 2001 [Wikip]. The newest WML version in active use is 1.3. WML provides navigational support, data input, hyperlinks, text and image presentation, and forms.

XHTML Mobile Profile (XHTML MP) is a hyper textual computer language standard designed specifically for mobile phones and other resource-constrained devices. The XHTML MP 1.2 DTS is the current recommendation finalized in March 2008. XHTML-MP is derived from XHTML Basic 1.0 by adding XHTML Modules and deprecating some modules.

Though those mobile markup languages are optimized for mobile browsing, it is time consuming and economically expensive to create a brand new mobile optimized Web site and maintain the consistency between the mobile version Web and the desktop one.

2.1.2 Automatic mobile Web site creating tools

Instead of creating a completely new mobile Web site with mobile specific markup languages, many mobile Web site creating tools have been proposed to create mobile Web sites with techniques designed for desktop version Websites and

these techniques are also supported by the mobile browsers. These tools can help Web designers create mobile optimized Web sites and make the task of creating mobile Web sites easily and fast.

Zinadoo [Zinad] proposed Web service to easily create and publish mobile Websites. During the creation, users can also color the Website, create a page that users can send emails from or request, create a guestbook page, a comments page and etc. Winksite [Winks] is a mobile Website builder that includes RSS-driven content deployment and mobile-tuned community features such as forum, chat, and polls. It does not require the user to install the software, and it does everything online. MobiSiteGalore [Mobis] allows designers build Websites that will work consistently across all mobile phones as it completely complies with the W3C's mobile Web standard [Wmw04]. This service is completely free. Zhang *et al.* [Zha05] proposed a grammatical approach for adaptive layouts by specifying a high level structure and spatial relations among multimedia objects through a graph grammar. This approach provides a powerful authoring tool, allowing designers to specify the adaptation of presentations when the viewing condition is dynamically changed.

2.2 Client Side Navigation Techniques

Many client side navigation techniques have been proposed in last decade. These navigation techniques intend to provide user-friendly browsing experience while keep the original layout of the Web content. These client-side navigation approaches include mini-map, scaling, zooming, thumbnail, slicing and etc.

[Sch09] conducted a multidimensional study on mobile browsing usage scenarios as well as the usability of mobile tailored compared to full Websites. The study results show that mobile optimized versions provide better user experience than full Websites. Bolt [Boltb] is a full desktop PC-style browser on all types of mobile phones. The Web content is never reformatted, repurposed, or removed. Split-screen mode features a rectangular magnifier floating over a zoomed out mini-map of the entire Web page on the top 2/3 of the screen and a magnified view of the content under the magnifier on the bottom 1/3 of the screen. The magnifier floating over the mini-map enables you to quickly find information and navigate Web sites with ease.

A thumbnail-based overview is another popular technique used together with zooming used in client-side navigation. Wobbrock et. al. [Wob02] proposed a system called WebThumb, which provides basic panning and zooming capabilities on the thumbnail of an HTML Web page. WebThumb provides a “pick mode” to support a user to quickly identify the information of interest and a “text mode” for reading a text paragraph while avoid zooming in to the thumbnail. Lam et. al. [Lam05] proposed a system called Summary Thumbnail. The Summary Thumbnail displays fragment of readable text in the region of each text block. The Summary Thumbnail preserves the original layout while allowing users to quickly differentiate text blocks with a similar look and identify the information of interest. Thumbnail images are also used many applications to adapt a Web page to fit the small screen [Bau04, Bjo99, Lam05, Mil03, Wob02]. A number of thumbnail images are created for different parts of a Web page and users can select one image

to view the specific part of the Web page. [Ara10] presented annotations on a Web page that provides functionality which automatically scrolls the page.

2.3 Web Content Restructuring

Web content restructuring approaches proposed the technologies that an adaptive layout reorganizes and reformats different topics within the original Web page to fit the small screen mobile device. According to W3C, content adaptation is defined as a process of selection, generation or modification to transforming contents to adapt to mobile device capabilities due to their limited computational power, small screen size and constrained keyboard functionality. Because HTML pages do not contain any semantic structure and also because mobile devices have limited display and computation power, the automatic Web content restructuring is very challenging. Many solutions have been proposed to address these challenges. They can be divided into two sub-categories: client-side restructuring and server-side restructuring. In these approaches, the generated adaptive mobile presentation is normally different from the original Web page.

2.3.1 Approaches of client-side restructuring

The client-side restructuring does not require a server to support the Web pages restructuring. The Web page restructuring job is done dynamically at the client side while the mobile users browse the Web sites. These approaches may consume a considerable amount of CPU power and memory of the mobile devices.

Bitstream ThunderHawk [Bistr] uses intelligent font resizing and tunes images to adapt the Web page. [Che03] organizes a Web page into a two level hierarchy with a thumbnail representation at the top level for providing a global view and index to a set of sub-pages at the bottom level for detailed information. [Gom11] uses a combination of multi-page organization and narrow-page organization to present the contents. [Yan03] employs the fractal theory to generate various summaries on different levels of the Web page based on demands of users. Minimap method [Vir06] shows pages in a modified layout by changing the Web page Cascading Style Sheets (CSS). The Gateway [Mac03] introduces a navigation model to minimize the effect of transformation volatility for users switching between different screen sizes and to fit the small screen. [Hat07] uses heuristic rules based on a set of HTML tags (such as *tr*, *p* and *ul*) to partition a Webpage into segmentations, and then adapts the Web page with an object title list from the segmentations, so that each individual segmentation can fit to small screens.

2.3.2 Approaches of server-side restructuring

In the server-side restructuring category, an application is running on a server, which fetches the Web page from a remote content server based on client requests, and then restructures the Web page to respond the client's requests.

Google provides a conversion utility through which users just need to enter the URL of their Website, check the option for 'No Images' if they do not want images to appear, and then click on 'GO' button. The corresponding mobile version

will pop up for mobile browsing. The Opera mobile browser [Opera] uses the software technique, called *Opera Turbo*, to restructure a Web page into a one-column layout with an adjustable width to fit the screen width. This approach always generates a long column where users must scroll up and down to find the desired contents. Google mobile pTHINC [Kim06] uses the server-side screen scaling approach and users can zoom into any desired areas of the page. The Skweezer Proxy [Skwee] reformats the layout of the content in a Web page to fit the small screen through the mobile content proxy site skweezer.net. [Par10] proposes Web contents trans-coding middleware that could provide automatic Web contents re-authoring for mobile devices.

2.3.3 Comparing with my approach

In this paper, I proposed an approach, which belongs to the Web content restructuring category. Compared with the approaches mentioned in previous sections, my approach has one significant difference:

- Multi-style presentation layouts. All of the existing approaches only provide one fixed presentation layout. One fixed presentation would not provide universal usability for diverse users and Web contents. My approach provides multi-style adaptive layouts. The multi-style layouts options provide users the freedom to choose layouts according to their preference and user experience, which eventually provide better user experience.

CHAPTER 3. EVALUATION OF MOBILE WEBSITES AND MOBILE WEBSITE GUIDELINES

Fixed presentation layout for a diverse range of users, Web sites and devices could not offer universal usability [Zha03]. To render an adaptive layout for a diverse range of users, I evaluated 18 existing mobile Websites in different categories such as news, online shopping, travel, social network and etc. I analyzed and classified existing adaptive layouts from two perspectives: navigation facility and information presentation [Ahm08]. From these popular mobile Websites, I was able to find the most popular adaptive layouts currently used in mobile browsing. In addition, this evaluation gives a solid basis to compare different approaches and provides a space on how to design an adaptive layout.

First, I noticed all 18 mobile Website use a table of contents to provide an overview of the Web page's contents. The difference between them lies in the information presented in the table. Some Websites use both image and textual label while others use textual label only. Secondly, through evaluating the layouts used by these mobile Websites to present the detailed information, I identified two common layouts:

- **New page:** The detailed information for a single topic is displayed in a separate page from the overview page.
- **Collapsing:** Detailed information for each topic is collapsed in the same page as the overview.

While most mobile Websites open a new page when a user clicks a link or a button of each topic, some Websites use the collapsing layout by showing the detailed information just under the topic when clicked.

The following sub-sections describe the evaluation from both navigation facility and information presentation perspectives in details [Ahm08]. Navigation facility focuses on the overview while the information presentation focuses on the presentation of detailed information.

3.1 Navigation Facility

Navigation facility refers to what facilities the mobile Website provides and how these facilities help mobile users quickly find the topics interested. More specifically, I analyzed the depth of navigation menu, the presentation of menu in each level and the search box position for each mobile Website (see table 1).

The column depth shows the depth of navigation menus of the Website (see table 1). For example, the homepage of CNN mobile site classifies the news stories into different sections. A dropdown list, which locates at the top of the homepage, lists all the news sections. After selecting any section, it lists several news stories under each section with the title of each news story. Therefore, the depth of the navigation menu in the CNN mobile site is 2 (see figure 1), while the depth of navigation menus of amazon.com is 3 (see figure 2). The first level navigation menu in amazon.com shows the popular merchandise categories such as Books and

Music. The second level shows all merchandise categories, and then the third level shows all merchandises in the selected categories.



Figure 1 2 Level Navigation Menu of cnn.com

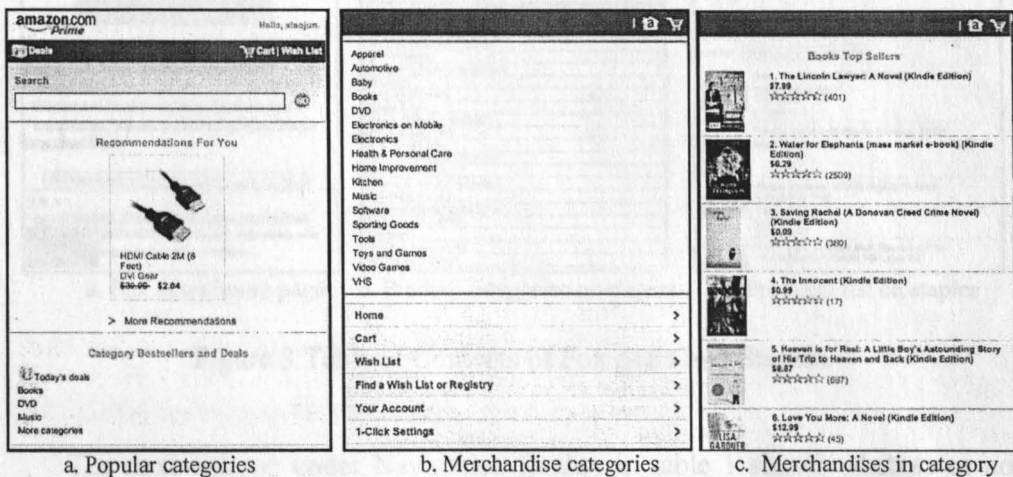


Figure 2 3 Level Navigation Menu of amazon.com

In addition to different depths in the navigation menu, the presentation of each navigational level is also different from one to another. Image, title and brief

description are the three primary information items displayed in the navigation menu, different combination of these three items are used in each navigation menu in different mobile Websites. The presentation on different levels shows the different combinations of information displayed in the navigation menu for each level. Fox news uses both the pictures and news title in the overview navigation table (see figure 3). Staples.com only uses labels for the different categories of products in the first three navigation levels while Image, production title and brief description are used in the last navigation level (See figure 3). Amazon.com mobile site only uses title in the first two levels, and uses title with image to list all the products in the third level.

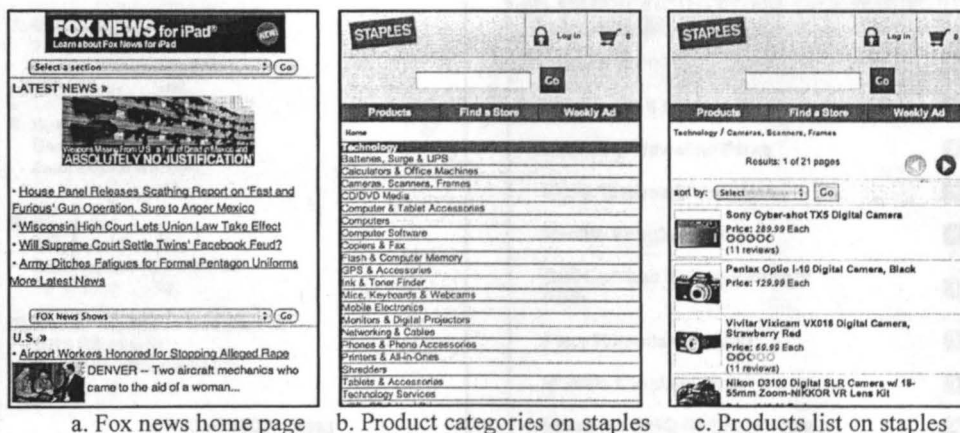


Figure 3 Table of Contents of Fox news and Staples

The next column under Navigation facility in table 1 shows whether or not the mobile sites use several pages in its navigation menu to list the menu items if there is a long list of the menu items. Some mobile sites such as

sites do not provide the search capability and no search box is presented in pages.

7. **Blowfish Sushi To Die For**
2170 Bryant St
San Francisco, CA 94110
☆☆☆☆☆ (106) 1.2 ml >

8. **Scoma's**
San Francisco, CA 94133
Exact location unknown
☆☆☆☆☆ (49) >

9. **Panta Rei**
431 Columbus Ave
San Francisco, CA 94133
☆☆☆☆☆ (62) 1.8 ml >

10. **Osaka Japanese Restaurant**
1923 Fillmore St
San Francisco, CA 94115
☆☆☆☆☆ (30) 1.2 ml >

1 - 10 of 4700 [Next](#)

[Home](#)
[Help](#)

 **near**
San Francisco, CA [change](#)

SPECIALTY PIZZAS		NAV
Domino's American Legends™		
Wisconsin 6 Cheese Pizza		
Honolulu Hawaiian Pizza		
Philly Cheese Steak Pizza		
Pacific Veggie Pizza		
Cali Chicken Bacon Ranch™ Pizza		
Fiery Hawaiian™ Pizza		
Buffalo Chicken Pizza		
Memphis BBQ Chicken Pizza		
Feast Pizzas		
Americas Favorite Feast®		
Bacon Cheeseburger Feast®		

Figure 4. Navigation lists of mynewpage.com and moodle.admnines.com

Website	Navigation Facility					
	Depth	Presentation of first level	Presentation of second level	Presentation of third level	One list / multi page list/tab	Search position
News:						
http://m.cnn.com	2	Title	Image, title	N/A	One list	Bottom
http://foxnews.mobi/	2	Title	Image, title	N/A	One list	N/A
http://wap.cbsnews.com/	2	Title	Image, title	N/A	List & tab	Top
http://m.aol.com	2	Title	Image, title	N/A	List & tab	Top
Online shopping:						
http://mobile.usablenet.com/mt/www.walgreens.com	4	Title	Title	Title	Multi page list	N/A
http://www.amazon.com	3	Title	Title	Image, title	Multi page list	Middle
http://www.staples.com	4	Title	Title	Title	One list	Top
http://m.sears.com	5	Title	Title	Title	Multi page list	Top
http://mobile.dominos.com	4	Title	Title	Title	One list	N/A
Travel:						
http://m.advantage.com	2	Title	Title	N/A	N/A	N/A
Job:						
http://mobile.monster.com	1	Title	N/A	N/A	Multi page list	New page
Social Network:						
http://mobile.twitter.com/	2	Title	Image, title	N/A	One list	New page
http://m.digg.com/	1	Title	N/A	N/A	Multi page list /tab	N/A
http://m.flickr.com/#/home	1	Image	N/A	N/A	Multi page list /tab	Bottom
Others:						
http://m.yellowpages.com/	3	Title	Image, title	Image, title, description	Multi page list	Top & Bottom
http://m.nfl.com	2	Title	Title	N/A	List/tab	N/A
http://m.youtube.com/	2	Title	Image, title	N/A	Multi page list	Top
http://www.google.com/m/finance	1	Image, title	N/A	N/A	List/tab	Top

Table 1 Evaluation of Navigation Facility

3.2 Information Presentation

With respect to the information presentation, I was focusing on the adaptive layouts employed in the detailed information pages. For example, the page shows the actual news stories in CNN mobile site or the page shows the product details in Amazon mobile site. I analyzed three aspects from the perspective of how users can interact with these pages: (1) is the information presented in one page or multiple-page? (2) What is the navigation method provided from one page to another? (3) How is further information related to this page presented? My evaluation of these three questions for these 18 mobile Websites are presented below (see table 2).

The Multi-/Single page column indicates the total number of pages to present one topic. I identified two layouts: multi-page and single page. For example, CNN mobile divides a long news story to several pages when the news story is too long to display in one single page while Staples mobile site always puts all the information about one product in one single page.

The navigation between different pages shows how users can switch from one page to another. I found that all mobile sites use the hyperlinks for users to navigate between different pages.

The progressive browsing shows how the mobile site presents further related information. Two different layouts are identified: New page and Collapsing. While most mobile Websites open a new page when a user clicks a link or a button to further browse the related information, some Websites use the collapsing layout.

For example, in YouTube mobile Website, when a user clicks the image of a video in the overview page, the user is redirected to a new page, which contains the video and user comments of that video. The user comments are collapsed by default and clicking the down arrow icon can expand the contents of user comments (see figure 5).

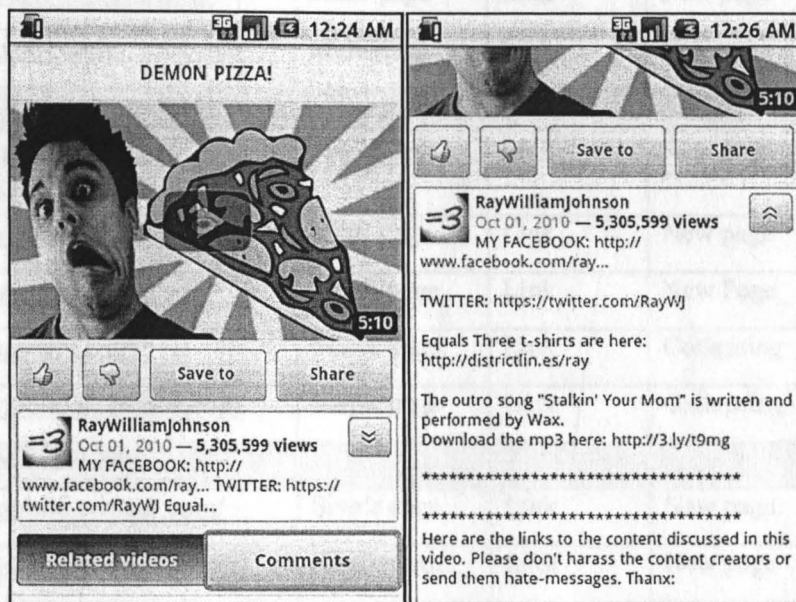


Figure 5 Collapsing Layout in YouTube.com

3.3 Evaluation Analysis

In summary, from the evaluation of these 18 mobile Websites, I noticed that both table of contents and progressing browsing are employed by most of them. This founding gives me a solid basis on how to design adaptive layouts on my research.

Website	Information presentation		
	Multi-page /Single page	Navigation between different pages	Progressive browsing
News:			
http://m.cnn.com	Multi-page	Link	Collapsing/new page
http://foxnews.mobi/	Multi-page	Link	New page
http://wap.cbsnews.com/	Multi-page	Link	New page
http://m.aol.com	Multi-page	Link	N/A
Online shopping:			
http://mobile.usablenet.com/m t/www.walgreens.com	Single page	Link	New page
http://www.amazon.con	Multi-page	Link	New page
http://www.staples.com	Single page	Link	New Page
http://m.sears.com	Single page	Link	Collapsing
http://mobile.dominos.com	Single page	Link	Collapsing
Social Networking:			
http://mobile.twitter.com/	Single page	Link	New page
http://m.flickr.com/#/home	Single page	Link	New page
http://m.advantage.com	Single page	Link	New page
http://mobile.monster.com	Single page	N/A	New page
Others:			
http://m.yellowpages.com/	Single page	Link	New page
http://www.iweathr.com/	N/A	Link	N/A
http://m.nfl.com	Multi-page	Link	N/A
http://m.youtube.com/	N/A	Link	Collapsing, new page
http://www.google.com/m/finance	Single page	Link	Collapsing

Table 2 Evaluation of Information Presentation

Table of contents provides the summary of the Web contents. All 18 mobile Websites use one or more tables of contents to provide an overview of the contents. The differences between these mobile Web sites are the information presented in the table of contents. The various combinations of image, title and brief description are presented in the table of contents in different Websites.

Progressive browsing provides the navigation method for users to browse the detailed information or related information. Through the evaluation of the layouts used in the detailed information page, I identified two different navigation methods: new page and collapsing. While most mobile Websites open a new page when a user clicks a link or a button for more information, other Websites use the collapsing layout by displaying the information right below current topic.

The next chapter describes how I adopted these two adaptive features in the prototype WebMini I developed.

3.4 Guidelines for Mobile Web Design & Development

In order to achieve high usability, I also studied the guidelines for mobile Web design & development. A mobile presentation should follow previous successful experiences, which are summarized as guideline documents. Those guidelines provide valuable guidance on producing a mobile Web page for both manual authoring and automatic adaptation. I classified the guidelines for mobile Web design & development in two categories: general guidelines and specific guidelines. The general guidelines are device independent while specific guidelines

are specialized to achieve the best user experience on one or several mobile devices, which are produced by a specific manufacturer. These guidelines provide me the basic ideas on how to develop a mobile Web application: MiniWeb, which is description in the next chapter.

One general guideline is the “Mobile Web Best Practices 1.0” [W3c08], a W3C Recommendation, which was developed by the Best Practices Working Group (BPWG) as part of the Mobile Web Initiative. It has been supported by many major names, such as Microsoft and Nokia. The principal objective is to improve the user experience of Web browsing from mobile devices. Mobile devices have many differences with desktops. The most significant differences are a small screen size and a different input method. The “Mobile Web Best Practices 1.0” provides a list of design principles to accommodate these two differences these general principles specify mobile Web design from five aspects: overall behavior, navigation and links, page layout and content, page definition and user inputs. W3C mobileOK checker is developed to automatically verify whether a mobile Web page observes Mobile Web Best Practices or not. In other words, it provides an objective evaluation on the usability of mobile Web pages. The generation of mobile Web pages, either manually or automatically through the approaches in Section 2, can be optimized to assure the observance of Mobile Web Best Practices

In addition to general guidelines, many mobile device manufacturers have published their own guideline documents that enhance user experience on one or

several specific devices. Apple provides the guidelines of designing graphics, media and visual effects specifically for iPhone [App09, App10]. Those apple specific guidelines provide enhanced user experience when a user is using the Safari Web browser on an iPhone or iPod Touch. In [Goo11], Google proposed the guidelines for Android- powered devices by using viewport metadata. Since Android is an open standard that has been used on various devices with different screen sizes and display resolutions, this guideline document especially discusses the scaling that provides the best visual presentation when a Web page is browsed on multiple types of screens. Nokia, which has the largest market share in mobile devices and smart phones worldwide, has been admitted to have the best support for developers. Hundreds of documents including guidelines for mobile Web development are available on its Website for developers. With the fast growth of touch screens in mobile devices, Nokia proposed the guideline document of *User Experience Checklist for Touch* [Nok10-1], especially considering touch-based applications. On the other hand, the Nokia *User Experience Checklist for Keypad* [Nok10-2] discusses the most important UI style issues when a keypad is used as the primary input method. The latest BlackBerry 6 has improved mobile browsing, such as supporting tabbed browsing and automatic network transport selection. The *BlackBerry 6 Browser Fundamental Guide* [Rim09] suggests that mobile developers should be aware of the physical components of a wireless browsing environment, including different version of the device software, screen size, input methods, wireless network and network gateway. This is especially important for mobile browsing since a user may be moving around different locations, which

cause a significant change on browsing environments. In addition, there are many other specific guidelines from different manufactures, e.g. *Developing Web Content for the HP webOS platform* [Hpd10], Opera's mobile Web optimization guide [Osa10], Sony Ericsson NetFront browser [Sem10] and etc. Specific guidelines are created by different organizations, and always focus on specific devices. Therefore, those guidelines may not be compatible with each other. For example, both Safari on iOS and Android Browser use the viewport to display a Web page, but they have different properties or behaviors. The default viewport width of an Android browser is 800 pixels while the default viewport width of Safari on iOS is 980 pixels. Since Safari is designed for iPhone which has a fixed screen size, a user can achieve the highest usability if the width of a mobile Web page is set as 980 pixels (i.e. the device-width) so that the scale is 1.0 in portrait orientation and the viewport is not resized when the user changes to landscape orientation. On the other hand, Android is used on various devices with different screen sizes, so it is impossible to have a page width that could be applicable to all devices.

CHAPTER 4. RESTRUCTURING

This chapter presents system design and the prototype development for the multi-style Web content restructuring mobile application. When we design mobile applications, we have to consider restrictions of mobile devices. For example, due to the small size of screen, only a small portion of information is visible at any time. Therefore, efficient navigation facility should be provided to users to enable quick access of the information [Jon99]. My system can restructure the information from normal desktop Web pages. Users are able to specify the presentation style according to their personal preferences. Furthermore, the adaptive layout provides an efficient navigation mechanism, which facilitates switching between different topics. Without losing generality, I developed a prototype, which supports different adaptive layouts based on user preferences.

Meanwhile, the survey in the previous chapter identifies the popular adaptive styles employed by most current mobile Websites. My system is trying to support all these popular adaptive styles such as collapsing, new page, and overview. A user can select the adaptive style according to his/her personal preferences and experiences with these different adaptive styles.

The prototype, named as MiniWeb, is developed under Android SDK 2.2 [Andr-2] and the ADT plug-in for eclipse. The rest of this chapter presents the system design and prototype implementation.

4.1 Multi-style Information Presentation

In my application, I used both adaptive layouts: table of contents and detailed information pages. The table of contents gives users an overview of a particular Web page by listing all records within the Web page. A record here refers to one product in online shopping Websites or one news story in news Websites. From the table of contents users can easily find the record they are interested in, and then jump to the detailed page, where the detailed information of the record is displayed. In summary, my application presents information in two levels: a table of contents and detailed information pages. It also provides customization options in both levels.

Users can change the information presented in the table of contents through a graphical user interface (see figure 6). According to their preference, users can select whether they want to show the image or text or the link of the record through a checkbox menu. This customization can help users to focus on the information they are interested in and remove all other information.

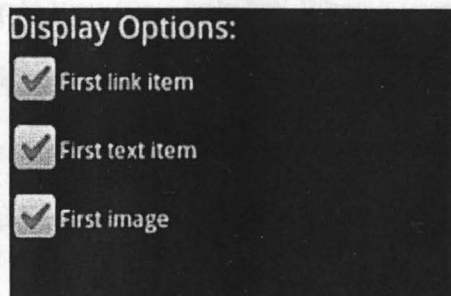


Figure 6 Display Option for Table of Contents

In detailed pages, the customization provides users with three options: Anchor Links, Single Page, and Collapsing (see figure 7). These three layout options are the summarization of adaptive layouts from 18 mobile Website I evaluated in the previous chapter. They are the most popular layouts employed in the mobile browsing, and provide users with the freedom to choose the most appropriate layouts according to their preferences.

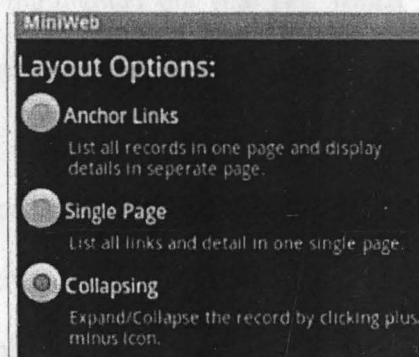


Figure 7 Layout Options for Detailed Page

The different layout options are explained below:

Anchor Links: The entire table of contents is presented in a List View. A user can scroll down or up to view the whole list. When a user finds any interesting record, he/she can click the record, and then a detailed information page corresponding to that record is shown in a different page where the user can slide the page to the left or right to view the previous or next record.

The left side of the picture shows the entire table of contents (see figure 8). For each record, it has a picture and a label for the product. On the right side of

each record, there is a right arrow icon. When a user clicks the right arrow icon, a new page will show up. We call this new page as a detailed page because it contains the detailed information of the record we just clicked. At the left top corner of the detailed page, there is a left arrow icon. Users will be redirected to the table of contents by clicking this icon. At the bottom of the detailed page, there are many small dots. The number of dots corresponds to the number of records in the table of contents and the highlighted dot (the one which is bigger than others) represents the position of current displayed record in the records list. A user is able to slide the page to the left in order to view the detailed page for the next record or slide the page to right in order to view the detailed page for the previous record. This navigation method gives users a shortcut to browser all records instead of selecting them from the table of contents.

Single Page: Both the table of contents and the detailed information are displayed in a single page, which could be a long, single column page. The entire table of contents is displayed at the top of this page, followed by the detailed information of each record. Users can scroll up and down to view the table of contents and the detailed information. When any record is clicked in the table of contents, the display automatically jumps to the location where the detailed information of that record is positioned. A user is also able to return to the table of contents by clicking an up arrow icon.

In the left side of the picture, each record in the table of contents has a down arrow icon (see figure 9). Clicking the down arrow icon will direct the user to the

position where the detailed information of this record is displayed as shown in the right side picture. Both the table of contents and the detailed information are displayed in the same page. A user has two choices to come back to the table of contents: by clicking the up arrow icon or scrolling up the records with a touch screen.



Figure 8 Anchor Link Layout Screenshot

Collapsing: The entire records summary is presented in a List View. By default the detailed information for each record is collapsed. By clicking the plus icon for any record, the detailed information for that record can be expanded right below the record. The plus icon is replaced with a minus icon. And the detailed

information can be collapsed again by clicking the minus icon. With this layout, users will not be redirected to a new place for detailed information.



Figure 9 Single Page Layout Screenshot

The left side picture shows each record in the table of contents has a plus icon, which indicates that the detailed information of the record is currently collapsed under the plus icon (see figure 10). The right side picture shows the detailed information for the second record is displayed and the plus icon for this record has been replaced with a minus icon.

4.2 Module Design

This section provides a detailed description of my prototype implementation. The overall architecture of the prototype is presented below (see figure 11).

The prototype encompasses three major modules: Website XML parser, Multi-style customization, and Page generation.



Figure 10 Collapsing Layout Screenshot

Website XML parser: An XML file is generated for each Web page through the pattern-based data extraction. The pattern-based data extraction is running on the server side, which accepts the request of Web page from the mobile application and then downloads Web contents from the URL specified by the request. It then extracts the Web content items based on the layout information and generates the XML file as the response of the mobile application's request. The XML file contains a spatial graph of informational objects in the Web page. By parsing the XML file, the mobile application gets the record list of the Web page as well as the details such as the title, links, images' URL and descriptions for each

record. A separate thread is used to download the images from original Website for each record. All of this information is stored in the memory for the information presentation module.

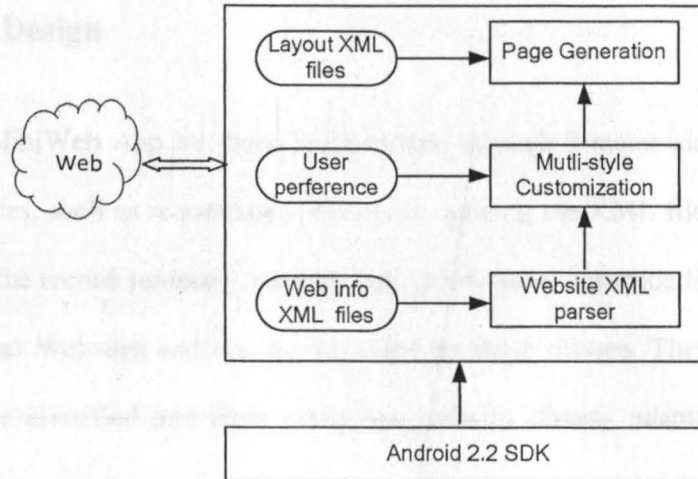


Figure 11 The Prototype Architecture

Multi-style Customization: This module provides the customization pages for users to choose their preferred layout. The prototype provides three layout options for users to choose. The three layout options are: Anchor Links, Single Page and Collapsing (see figure 9). In addition to the three layout options, the mobile application also provides three display options for the table of contents (see figure 10). The users' customization will be stored and used by the page generation module until they change their layout.

Page Generation: The page generation module is responsible for the information presentation according to the layout currently chosen by the user. Android framework uses XML file to define the layout structure that appears to the

user. I defined one XML for each layout mentioned in previous sections. Therefore, if a user selects any layout option the corresponding XML is automatically loaded and the page is displayed to the user.

4.3 Class Design

The MiniWeb App has been implemented through 9 major classes. Various functionalities, such as requesting a Web page, parsing the XML file a Web page, and listing the record summary, opening the option pages, downloading the images from original Websites and etc. are provided by these classes. These classes can generally be classified into three categories: activity classes, adapter classes and background task classes.

Activity Classes: In Android framework, an activity is an application component that provides a screen with which users can interact to complete different tasks, such as dial the phone, take a photo, send an email, or view a map [Andr-1]. An activity can only have one screen. When an application has multi screens, it needs to create multi activity classes for each screen. The MiniWeb application has 4 activity classes as listed below:

- **GalleryProduct:** The activity class displays the table of contents with only images.
- **MiniWeb:** This activity class displays the table of contents with titles, brief description and/or images.

- **PreferencesAct:** This activity class displays the multi-style layout options screen.
- **WebSitesAct:** This activity class displays the Websites list screen.

The MiniWeb Class is the main activity class of the prototype. It is the entry point and provides the first screen of the prototype. Its responsibility includes displaying the Web content with current selected presentation layout, opening other screens upon the user's request, managing backend threads such as downing the images for each record.

Adapter classes: An adapter object acts as a bridge as between the adapter view and the underlying data for that view. The adapter class separates the view and data and makes it possible to provide multi views with same underlying data. MiniWeb has three adapter classes as listed below:

- **GridViewAdapter:** This adapter is used by the grid view in MiniWeb class for displaying the content of table.
- **ListPicAdapter:** This adapter is used by the list view in MiniWeb class for displaying the detailed information for each record when a single page layout is selected.
- **ProductViewAdapter:** This adapter is used by the GalleryProduct class for displaying the picture of each record.

Background Task Class: In order to provide quick response to users' actions, some tasks need to be put in background threads. In MiniWeb application, there are

two tasks, which need to be implemented as background threads. The first one is the task of downloading and parsing Web content xml file, which are done through `parseWebsiteXML` class. The `parseWebsiteXML` class is extended from `AsyncTask` class. `AsyncTask` has 4 steps to run (see figure 12). The first step of `onPreExecute` is normally used to setup the task, such as initiating the variables; the second step `doInBackground` is invoked on the background thread which performs background computation such as downloading the actual XML file as well as parsing it; the third step `onProgressUpdate` is used to display the progress of the background task to UI thread. The last step `onPostExecute` is invoked after the background task has finished, and it updates the results to UI.

Another background task class is the `PhotoLoader` class, which is responsible for downloading the images from original Websites. The `PhotoLoader` class is extended from `Thread` class, and keeps running on background. When the main thread has images to display, a task of downloading the image is put into a task queue. The `PhotoLoader` thread keeps checking the task queue and downloads the images when a download task is found in the task queue. The activity diagram for `PhotoLoader` thread is presented below (see figure 13).

4.4 Class Diagram

The class diagram of the MiniWeb application is presented below (see figure 14). It contains nine major classes used by MiniWeb project. The functionality of each class is described below:

- **PreferencesAct:** The PreferencesAct class displays the user preference page. It is displayed when a user selects preference menu item. Users can select their preferred adaptive layout in this page.

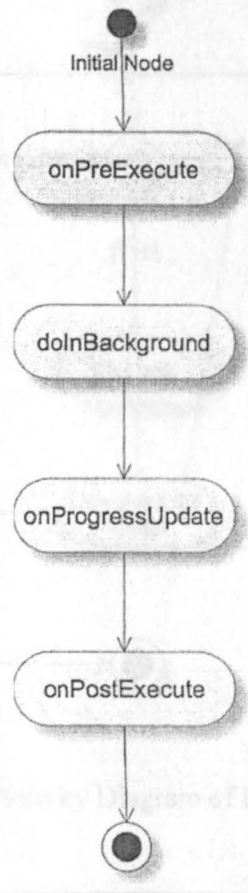


Figure 12 The Activity Diagram of AsyncTask

- **ImageLoader:** The ImageLoader class is responsible for images management including image downloading and image cache. It has its own thread to monitor the image download queue and downloads the image from original Website.

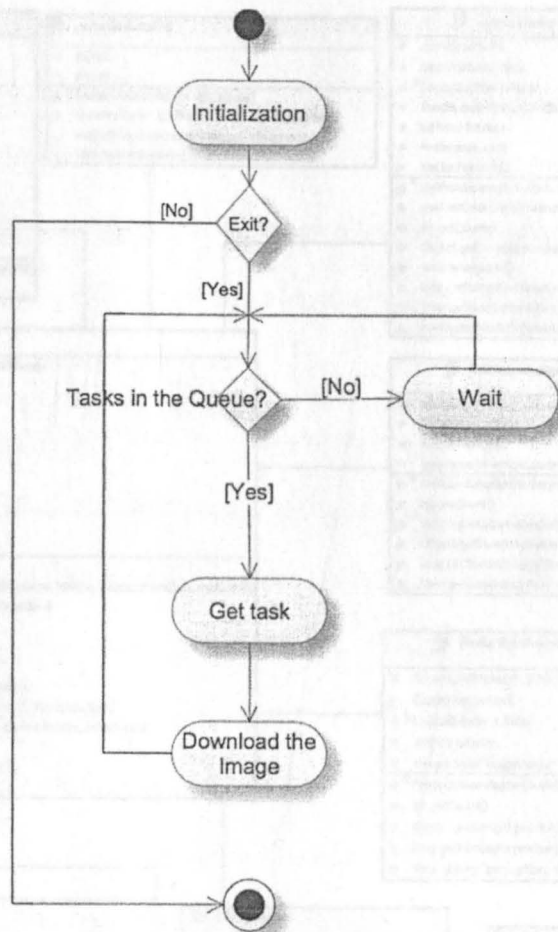


Figure 13 The Activity Diagram of PhotoLoader Class

- **MiniWeb:** The MiniWeb class is the center class of the MiniWeb. It is also the main entry of this application. It displays the table of contents of the Web pages. It wraps the application startup and shutdown procedures and calls other classes & activities when necessary.

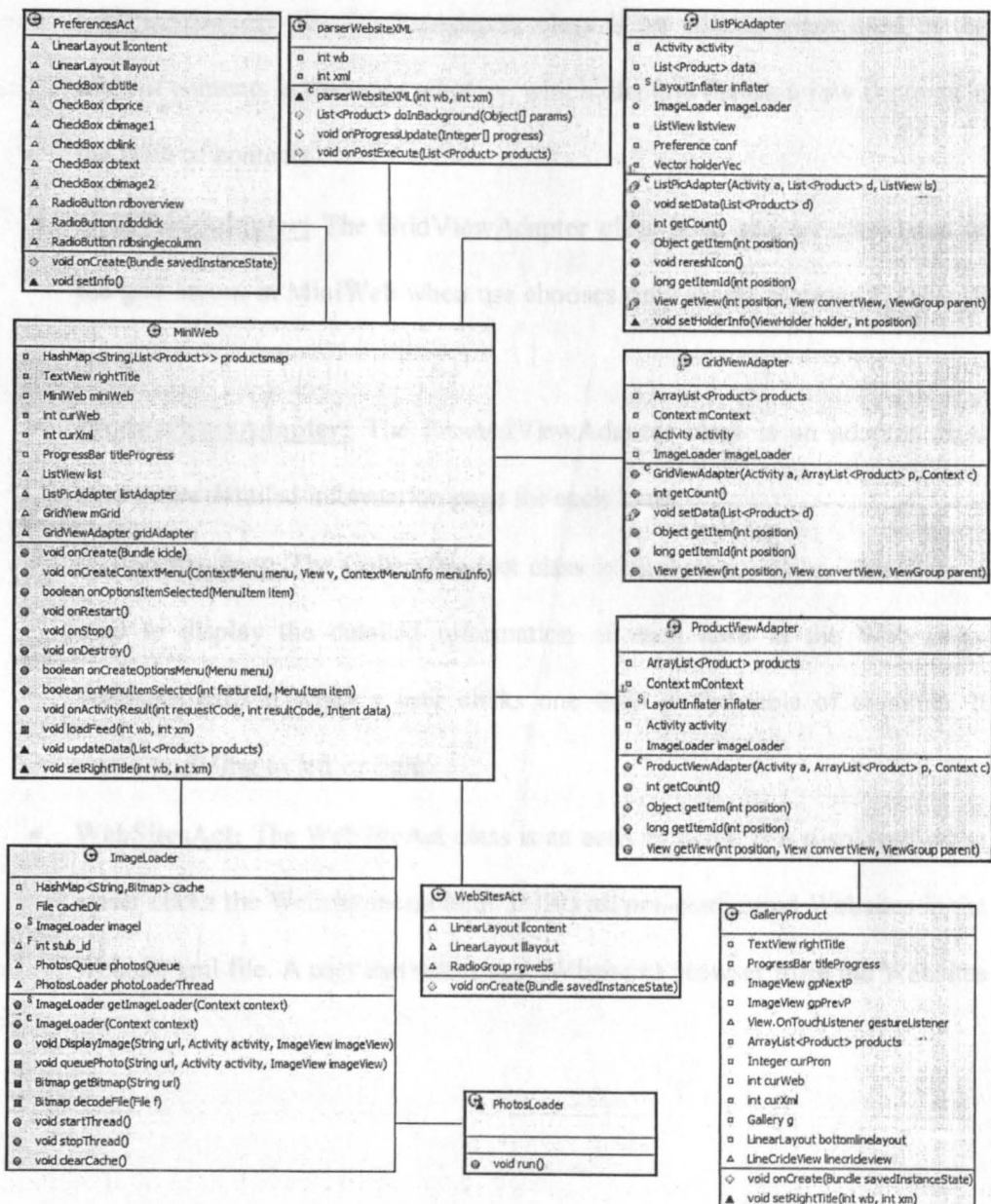


Figure 14 Class Diagram

- ParserWebSiteXML:** The ParserWebSiteXML class parsers the Websites list xml file when the application starts up. The websites list will be stored in a list and displayed when a user clicks the Website menu item.

- **ListPicAdapter:** The ListPicAdapter class is an adapter class used in the table of contents in the main window, which displays the each row contents in the table of contents.
- **GridViewAdapter:** The GridViewAdapter class is an adapter class used in the grid layout in MiniWeb when use chooses only shows pictures in table of contents.
- **ProductViewAdapter:** The ProductViewAdapter class is an adapter class used in the detailed information page for each item.
- **GalleryProduct:** The GalleryProduct class is an Activity class. This class is used to display the detailed information of each item in the Web page. WebMini calls it when a user clicks one item in the table of contents. It supports sliding to left or right.
- **WebSitesAct:** The WebSiteAct class is an activity class. It is displayed when a user clicks the Website menu item. It lists all pre-configured Websites in the Website xml file. A user can select any Website to browser from the Websites list.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

Over the past several years we have witnessed an enormous surge in the usage of mobile devices, such as PDAs, cellular (smart) phones, and iPod/iPad. More and more people are using these mobile devices to browse the Web to get information, entertainment, and knowledge. Web browsing using portable mobile wireless devices poses a new challenge to Web page presentation on the small screen of mobile devices. As presented in previous chapters of this paper, existing researches have overcome a lot of the challenges of this problem. However, intelligently and effectively adapting an existing Web page designed for desktop to a portable device with much smaller screen is still an open research problem.

To solve the aforementioned problem, this paper presents a novel approach for automatically restructuring a full Web page to fit for mobile browsing. I studied the current popular layouts employed in mobile Websites. Through this study, I obtained a solid understanding about the state of the art, and it gives me insight on how to design an adaptive layout. Then I proposed a novel multi-style Web content restructuring approach for efficient mobile browsing. I designed the system in a way that it can present the information in multiple styles for diverse users. The customization capability enables users to choose the presentation style that fits the users' needs on mobile devices. My approach provides an efficient navigation facility and much better usability comparing to current Web mobile Web browsers.

I have implemented a prototype to evaluate the efficiency of my approach. The prototype provides customization capabilities to the users. The prototype provides good user experiences on browsing Web pages and compared with normal phone browsers.

5.2 Future Work

Important problems in adaptive layout presentation of portable mobile devices remain to be solved. I identify several limitations of my work and research directions for future work.

I will continue the research on developing more adaptive layouts to cater to the tastes of the diverse users while improving the usability. The user interface of Web content presentation on mobile device could be extended and optimized based on users' suggestions and preferences. More studies could be done to add more presentation options to the users. Meanwhile, I will also continue to research on how to present the Web contents under different categories to generate efficient and user-friendly adaptive presentations. More evaluation will be carried out to different user groups, such as students, businesspersons, and seniors to get feedback from different user groups.

REFERENCES

- [Ahm08] Ahmadi, H. (2008), Efficient Web Browsing on Small Screen, master thesis.
- [App09] Apple. (2009). Safari on iPhone Graphics, Media and Visual Effects Coding How To's. Retrieved from <http://developer.apple.com/library/safari/#codinghowtos/Mobile/GraphicsMediaAndVisualEffects/index.html>
- [App10] Apple. (2010). Safari Web Content Guide. Retrieved from http://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariWebContent/Introduction/Introduction.html#//apple_ref/doc/uid/TP40002051
- [Andr-1] Activity in Android Framework. (2011). Retrieved from <http://developer.android.com/guide/topics/fundamentals/activities.html>
- [Andr-2] Android SDK document. (2011). Retrieved from <http://developer.android.com/sdk/index.html>
- [Ara10] Arase , Y., Hara,T., Uemukai,T., Nishio, S. (2010) Annotation and Auto-Scrolling for Web Page Overview in Mobile Web Browsing. *International Journal of Handheld Computing Research*, vol 1(4), p.63-80
- [Bau04] Baudisch, P., Xie, X. Wang, C., Ma, W. (2004). Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content. *Proc. UIST'04*, p.91-94

- [Bistr] A Mobile Web Browser from Bitstream. Retrieved from <http://www.bitstream.com/wireless>.
- [Bjo99] Bjork, S., Bertan, I., Danielesson, R., and Karlgren, J. WEST (1999). A Web Browser for small Terminals. *Proc. UIST'99*, p.187-196
- [Che03] Chen, Y., Ma, W., and Zhang, H. (2003). Detecting web page structure for adaptive viewing on small form factor devices. *Proc. WWW'03*.
- [Gom11] Google mobile. (2011). Retrieved from <http://www.google.com/mobile>.
- [Goo11] Google. (2011). Targeting Screens from Web Apps. Retrieved from <http://developer.android.com/guide/webapps/targeting.html>
- [Hat07] Hattori, G., oashi, K., Matsumoto, K. and ugaya F. (2007). Robust web page segmentation for mobile terminal using content distances and page layout information. *Proc. WWW'07*, p.361-370.
- [Hpd10] Hewlett-Packard Development Company. (2010). Developing Web Content for the HP webOS platform. Retrieved from http://developer.palm.com/index.php?option=com_content&view=article&id=1574
- [Jon99] Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K. and Buchanan, G. (1999). Improving Web interaction on Small Displays. *Proc. WWW'99*, p.51-59.
- [Kim06] KIM, J., BARATTO, R. A., AND NIEH, J. (2006). pTHINC: A thin client architecture for mobile wireless web. *Proc. WWW'07*, p.143-152.
- [Lam05] Lam, H., Baudisch, P.(2005) Summary Thumbnails: Readable Overviews for Small Screen Web Browsers. *Proc. CHI'05*, p.681-690
- [Mil03] Milic-Frayling, N., Sommerer, R., Rodden, K., and Blackwell (2003). A. F. Web Viewing and Search for Mobile Devices. *Proc. WWW '03*, poster.

[Mac03] Mackay, B. (2003). The Gateway: A Navigation technique for Migrating to Small Screens. Proc. *CHI '03*, p.684-685.

[Mobis] mobiSiteGalore: a mobile website publisher. (2011). Retrieved from <http://www.mobisitegalore.com>

[Nok10-1] Nokia. (2010). User Experience Checklist for Touch. Retrieved from http://www.forum.nokia.com/dp?uri=http%3A%2F%2Fsw.nokia.com%2Fid%2F115f45ba-d765-4f86-8eeb-df396eb6e3bd%2FUX_Checklist_for_Touch_v1_0_en.pdf

[Nok10-2] Nokia. (2010). User Experience Checklist for Keypad. Retrieved from http://www.forum.nokia.com/dp?uri=http%3A%2F%2Fsw.nokia.com%2Fid%2F36d56f57-044d-4d4d-aece-c4a6396edf7f%2FUX_Checklist_for_Keypad_v1_0_en.pdf

[Osa10] Opera Software ASA. (2010). The Opera's mobile Web optimization guide. Retrieved from <http://dev.opera.com/articles/view/the-mobile-web-optimization-guide/>

[Opera] Mobile browsers from Opera. (2011). Retrieved from <http://www.opera.com/>

[Par10] Park, S., Chang, H., Jang, B., Ahn, H. Choi, E. (2010). A Design of the Transcoding Middleware for the Mobile Browsing Service. *Communication And Networking*, vol. 120, p.198-204.

[Rim09] Research In Motion Limited. (2009). BlackBerry 6 Browser Fundamental Guide. Retrieved from http://docs.blackberry.com/en/developers/deliverables/5687/BlackBerry_Browser-4.7.0-US.pdf

- [Sch09] Schemid, G., Seidl, M., Temper, K. (2009). Mobile phone web browsing: a study on usage and usability of the mobile web. *MobileHCI'09*, No. 70.
- [Sem10] Sony Ericsson Mobile Communications AB. (2010). Sony Ericsson NetFront browser. Retrieved from http://developer.sonyericsson.com/cws/download/1/708/159/1277364112/DW-88004-dg_web_browser_nf3_r31a.pdf
- [Skwee] Skweezer: A mobile web site publisher (2011). Retrieved from <http://www.greenlightwireless.net/>
- [Vir06] Virpi, R., Popescu, A., Koivisto, A. Vartiainen, E. Minimap. (2006). A Web Page Visualization Method for Mobile Phones. *Proc. SIGCHI'06*, p.35-44.
- [W3c97] The World Wide Web Consortium. (1997). Handheld Device Markup Language Specification. Retrieved from <http://www.w3.org/TR/NOTE-Submission-HDML-spec>
- [W3c08] The World Wide Web Consortium. (2008). Mobile Web Best Practices 1.0. Retrieved from <http://www.w3.org/TR/mobile-bp/>.
- [Wikip] WAP protocol. (2011) Retrieved from http://en.wikipedia.org/wiki/Wireless_Application_Protocol
- [Winks] Mobile website creating service from winksite. Retrieved from <http://www.winksite.com>
- [Wob02] Wobbrock, J., Forlizzi, J., Hudson, S., Myers, B. WebThumb. (2002). Interaction techniques for small-screen browsers. *Proc. UIST '02*, p.205-208
- [Wmw04] W3C Mobile Web Standards, Retrieved from <http://www.w3.org/2004/02/Mobile.html>

[Yan03] Yang, C.C., Wang, F.L. (2003). Fractal summarization for mobile devices to access large documents on the web. *Proc.WWW'03*, p.215 - 224.

[Zinad] Mobile website creating service from Zinadoo. Retrieved from <http://www.zinadoo.com/>

[Zha05] K. Zhang, J. Kong, M.K. Qiu, and G.L. Song. (2005). Multimedia Layout Adaptation Through Grammatical Specifications. *ACM/Springer Multimedia Systems*, 10(3), p.245-260.

[Zha03] Zhang, D. (2003). Delivery of personalized and adaptive content to mobile devices: a framework and enabling technology. *Communications of AIS*, vol.12, p.183-202.